Memories for Virtual AI Characters

Fabian Landwehr
ETH Zürich
fabian.landwehr@inf.ethz.ch

Erika Varis Doggett
erikavaris@gmail.com

Romann M. Weber
romann.weber@gmail.com

Abstract

In this paper, we present a system for augmenting virtual AI characters with long-term memory, enabling them to remember facts about themselves, their world, and past experiences. We propose a memory-creation pipeline that converts raw text into condensed memories and a memory-retrieval system that utilizes these memories to generate character responses. Using a fact-checking pipeline based on GPT-4 (OpenAI), our evaluation demonstrates that the character responses are grounded in the retrieved memories and maintain factual accuracy. We discuss the implications of our system for creating engaging and consistent virtual characters and highlight areas for future research, including large language model (LLM) guardrailing and virtual character personality development.

1 Introduction

Memory is arguably one of the keys to personality, influencing how we think and respond to external stimuli (Thorne). It is also pivotal for maintaining a coherent and engaging conversation (Baddeley). Such characteristics of memory hold considerable implications for the field of natural language processing (NLP), particularly for AI chatbots, which most recently have emerged as a significant form of human-computer interaction (Brown et al.; Adamopoulou and Moussiades; Chen et al.).

Recently, chat applications utilizing large language models (LLMs) have incorporated a type of short-term memory into their design, namely the chat history, which is included in the context window of the LLM (Zaib et al.). However, the integration of long-term memory into these systems is not yet standard. One reason is that without supporting systems, long-term memory can only be achieved by frequently retraining the underlying LLM, which is impractical due to prohibitive computational costs and associated emissions (Patterson et al.; Sharir et al.).

Most AI chatbots today are designed to function as assistants, parsing user inputs as commands or providing answers to queries, often relying on the support of information-retrieval systems (Adamopoulou and Moussiades).[1] However, envisioning a slight shift in this objective—using LLMs to generate responses for virtual AI characters—presents new challenges and opportunities.

The incorporation of long-term memory becomes critical in this context. To ensure that a virtual character appears human-like to a user, it must accurately remember facts about its universe and recall past interactions. Achieving this could elevate the user experience to a new level, creating a dynamic, immersive experience that adapts to the user (Madotto et al.; Thue et al.).

This paper presents a system to augment virtual AI characters with persistent, long-term memories. At a high level, we achieve this by generating character responses with an LLM, augmenting the prompts with relevant *memories*—condensed versions of character experiences, knowledge, or descriptions.[2] Figure 1 shows how memories can influence character responses.

Our exploration spans relevant related work, and we provide a detailed explanation of our system, encompassing retrieval systems, potential LLM steering, memory creation, and forgetting mechanisms. We evaluate our system based on two characters: *Sherlock Holmes*, a well-known character present in the LLM's training data and implicitly represented by the model's weights, and *Amina Osei*, a newly created character with no such representation. The evaluation focuses on the alignment between the character responses and the retrieved memories and uses a semi-automated fact-checking pipeline that leverages GPT-4.

[1] Examples include Apple's Siri, Google's Assistant, and Amazon's Alexa.

[2] We use OpenAI's `gpt-3.5-turbo` as the underlying LLM (https://platform.openai.com/docs/models/gpt-3-5).

Proceedings of the 16th International Natural Language Generation Conference, pages 237–252
September 11–15, 2023. ©2023 Association for Computational Linguistics

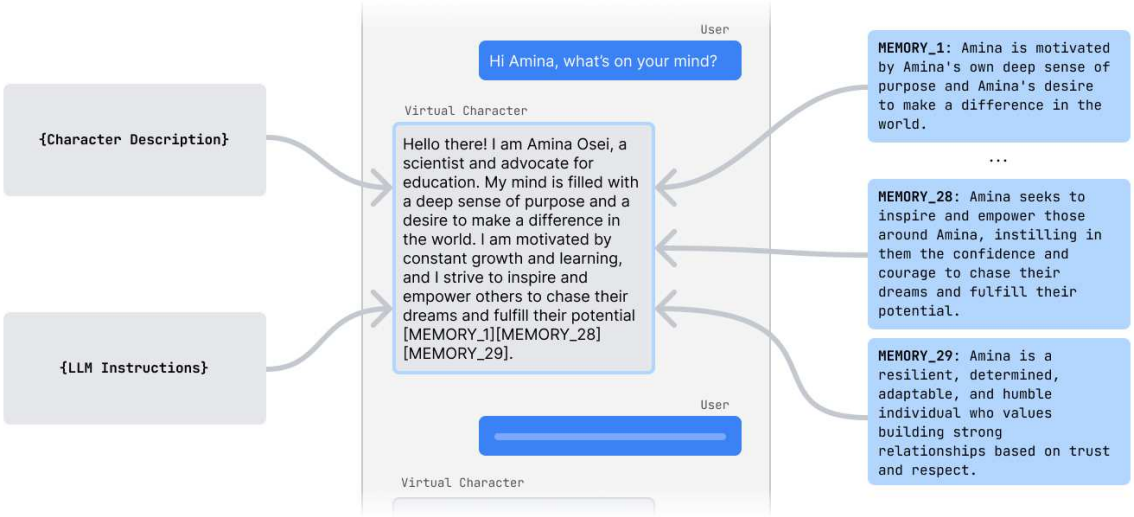
{Character Description}
{LLM Instructions}
User
Hi Amina, what's on your mind?
Virtual Character
Hello there! I am Amina Osei, a scientist and advocate for education. My mind is filled with a deep sense of purpose and a desire to make a difference in the world. I am motivated by constant growth and learning, and I strive to inspire and empower others to chase their dreams and fulfill their potential [MEMORY_1][MEMORY_28] [MEMORY_29].
User
Virtual Character
MEMORY_1: Amina is motivated by Amina's own deep sense of purpose and Amina's desire to make a difference in the world.
...
MEMORY_28: Amina seeks to inspire and empower those around Amina, instilling in them the confidence and courage to chase their dreams and fulfill their potential.
MEMORY_29: Amina is a resilient, determined, adaptable, and humble individual who values building strong relationships based on trust and respect.

Figure 1: *Memories for virtual AI characters:* This figure illustrates how virtual character responses are generated. When answering, characters have access to a list of relevant memories, a description of themselves, and the chat history. Section 3 explains the proposed system in detail.

This paper makes several contributions. First, it demonstrates how LLMs can be substantially enhanced by integrating them into a more extensive system. Second, it introduces a process of creating memories from unstructured text, simplifying the character-creation process. Third, it outlines partitioning memories into multiple "buckets," offering greater flexibility and potential for dynamic story progression. Fourth, it proposes a new query-generation approach resulting in targeted retrieval of relevant memories. Furthermore, it presents a forgetting model that mimics human memory patterns and allows customization of character memory traits.

2 Related Work

The proliferation of natural language (NL) interfaces in computer systems, including systems such as Siri, Alexa, and Google Assistant, has opened new avenues for intuitive interaction. Typically, these interfaces focus on question answering, command parsing, and named entity recognition, assisting users in a variety of tasks (Hussain et al.; Zemčík).

Another strand of research centers around developing conversational chatbots capable of realistic user interactions. Applications range from in-game non-player-characters (NPCs) (Park et al.) to virtual companions, as popularized in various science-fiction movies.[3] Our work intersects these two streams of research, aiming to create virtual characters capable of recalling factual information about themselves and their past experiences, effectively giving them long-term memory.

2.1 Character-Simulation Approaches

Traditional Methods

Historically, character simulation and chatbots relied on scripted dialogues or dialogue trees created by human designers (Hussain et al.; Zemčík). These systems employed key-phrase insertion for personalization (Weizenbaum). However, their lack of variability was a limitation, as no dialogue tree can accommodate all potential user topics. Consequently, interactions with these chatbots can feel robotic, as they cannot improvise.

Recent Approaches

Recent efforts such as OpenAI's ChatGPT and Microsoft's Bing have leveraged generative language models to generate responses.[4] However, issues such as excessive variability and hallucinations hinder their use for persona-based characters (Greshake et al.; Welleck et al.; Ji et al.). These issues have led to the exploration of knowledge-grounding techniques for generative language models, which we discuss further in section 2.2.

Most recently, the concept of generative agents

[3]Popular movies featuring AI companions include *Her* (2013) and *Blade Runner 2049* (2017). Examples of businesses that offer virtual companions include replika.com and myanima.ai.

[4]ChatGPT: https://openai.com/blog/chatgpt, Bing: https://www.microsoft.com/en-us/edge/features/bing-chat

has gained traction (Park et al.). Their focus is autonomously making decisions and planning actions in a virtual world based on past experiences and conversations. Some projects also attempt to let agents interact with the real world (Sig; Nakajima). Those approaches use LLMs in an infinite loop, where the output of one call can influence the input of the next. Similarly, our work uses LLMs as part of a broader software system to guide and limit the generated outputs. However, we emphasize artist-crafted virtual characters that engage in user conversations rather than autonomous agents.

2.2 Question-Answering (QA) Systems and Search Engines

This branch involves systems designed to provide factual answers to user queries. While not solely focused on question answering, our project integrates relevant character and context information into responses.

Transformer-based (Vaswani et al.) LLMs are now used in search engines such as Bing. Instead of encoding all information in the model's weights, an existing information-retrieval system is queried, and the results are taken as inputs for the LLM. Similarly, LLM wrapper libraries such as Langchain also provide implementations for external databases and memory to increase retrieval capabilities.[5] This splitting of knowledge and language capabilities simplifies the process of updating information (Wang et al.; Levine et al.).

We utilize a similar approach by storing character-relevant facts in a vector search database. Simultaneously, we differentiate our work by making memories human-like. We apply them to virtual characters, adding more sophisticated memory formation and retrieval mechanisms, including a forgetting function.

According to Lan et al., LLMs are used in question-answering tasks via *parse-then-execute* or *retrieval-and-rank* strategies. The former involves comprehension and query formulation, and the latter directly uses user input to query a knowledge base and subsequently filters the results. We combine these strategies in our system.

Alternative methods for knowledge grounding of language models exist, which include:

- Replacing some attention-based feed-forward networks with updateable context vectors (Sukhbaatar et al.).
- Nearest-neighbor language models to bias language generation toward specific datasets, simplifying updates (Khandelwal et al.).
- Retrieving text chunks similar to the input and using cross-attention for prediction, offloading LLM storage and facilitating updates (Borgeaud et al.).

While custom architectures offer the advantage of finer-grained control, we use general-purpose LLMs to be able to use powerful, pre-trained SOTA models, facilitating rapid prototyping.

3 Proposed System

The overarching idea of the proposed system is to generate character responses by prompting a general-purpose LLM with a dynamically created prompt based on the chat context and relevant memories. This section explains how different system components work together to create this prompt and how interactions are distilled into memories.

Figure 2 explains the system on a high level. A detailed description of the system's most central components and stages follows. Note: For all of our core system components described in this section, we use `gpt-3.5-turbo` (Brown et al.) as the base LLM.[6]

3.1 Chat History

The chat history contains all messages of the current conversation which still need to be converted to memories. This chat history is used throughout the system as a form of short-term memory and serves as the immediate context for the last user message.

3.2 Query Creator

Once the chat history has been updated with the most recent user message, the query creator generates a search query which is used to retrieve relevant memories. Appendix A.4 shows the prompt to create a search query. We noticed that using a low temperature was essential to making the query generation predictable.[7]

[5] https://python.langchain.com/docs/modules/memory/

[6] https://platform.openai.com/docs/models/gpt-3-5

[7] *Temperature* is a hyperparameter in LLMs controlling output randomness. Lower values yield more deterministic outputs focused on the highest probability predictions, while higher values promote diversity. A temperature of 1.0 maintains the model's original output distribution.

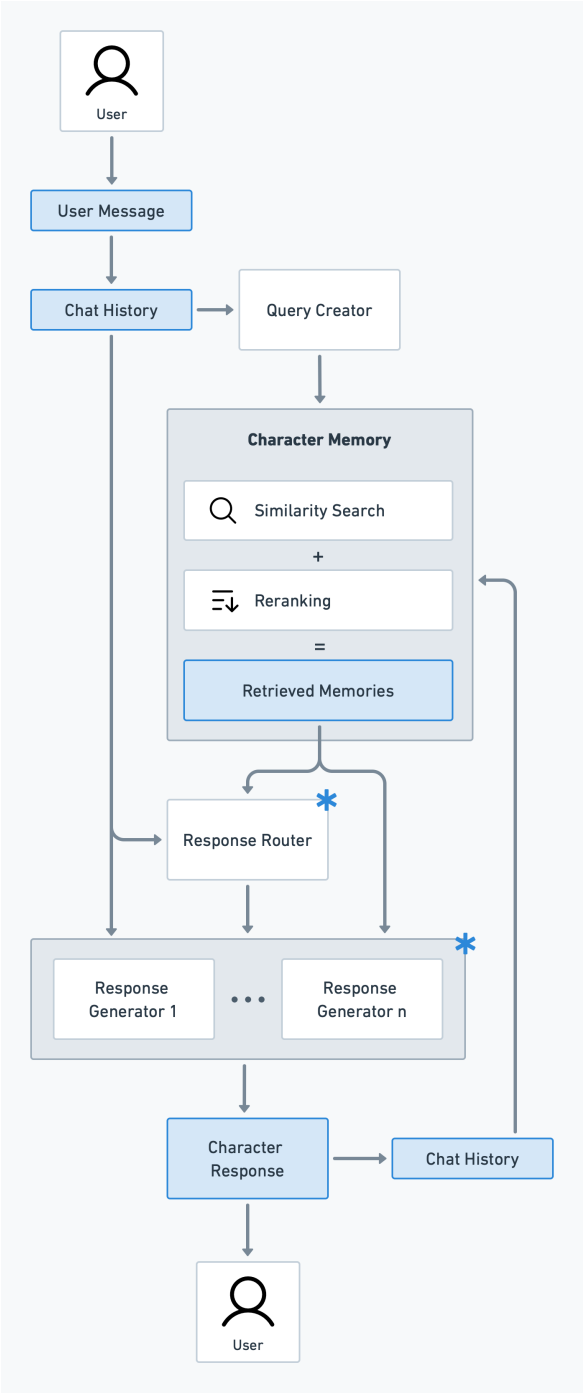

Figure 2: *Overview of the System Architecture:* When a user sends a message to a virtual character, a query for needed information is generated. Using this query, relevant memories are retrieved by doing a similarity search and re-ranking the results. Finally, an LLM is used to generate the character response. Interactions are later synthesized into new memories. For details, see section 3. The response routing (*) is part of our theoretical framework but not implemented in the current system; see section 5.

We retrieve memories based on similarity to the search query instead of similarity to the user input because, in some cases, the user input contains insufficient information. For example, if the latest user message was "Who is she?" then without knowledge of previous messages, it is unclear to whom the pronoun "she" refers. As a result, retrieved memories would likely be irrelevant.

3.3 Retrieving Memories

The character's memory consists of multiple knowledge sources. Each knowledge source conceptually maps to one source text. For example, one knowledge source could correspond to one Wikipedia article, another could correspond to a character description written by a novelist, and yet another could correspond to past conversations with the current user.

Each available knowledge source is queried using the created query to retrieve relevant memories. More precisely, a kNN search is performed on each knowledge source, using an embedding of the query and embeddings of the memory contents. This way, the memories most semantically similar to the query are found. For simplicity, we are using `text-embedding-ada-002` as an embedding function.[8] Future work should investigate if other embedding functions would yield better results.

A relatively high number of memories are retrieved from each knowledge source, and they will be re-ranked in the next step.

3.4 Re-ranking Memories

This step aims to prioritize and filter the high number of retrieved memories from different knowledge sources to fit into the context window of the LLM. The input to the re-ranking step is multiple lists of memories, one list per knowledge source, each sorted in descending order by similarity to the query. The lists are combined and sorted by a ranking score assigned to each retrieved memory. The top n memories with the highest scores are returned, where n is a modifiable parameter.

The memory scoring function allows for a high degree of customization. Multiple factors can be considered and combined into a score:

- *Similarity:* The similarity between the search query and the retrieved memory.

[8] https://openai.com/blog/new-and-improved-embedding-model

- *Retention:* How "strong" the memory is, i.e., the inverse of how much it was already forgotten, as explained in section 3.8.
- *Possible further factors:* Memory importance, trust in the corresponding knowledge source, user preferences, current emotional state of the character, progression of some underlying story, and more.

The scoring function is a weighted sum with with k factors x_i and associated weights w_i:

$$R \cdot \sum_{i=1}^{k} w_i \cdot x_i$$

Notably, the retention value R (see section 3.8) must not be part of the weighted sum, as a virtual character should not be able to recall a forgotten memory ($R = 0$), irrespective of the other factors.

We compute a ranking score for each memory using this formula. Finally, we return the n memories with the highest score.

3.5 Character Response Generation

As we explain in section 5, there could potentially be multiple strategies to create character responses. Here, we present the "base" strategy, used when the retrieved memories are relevant.

In this case, the character response is generated using the LLM with a dynamic prompt. The prompt contains the following:

- *Instructions:* The instructions tell the LLM to simulate a virtual character and to answer the last user message. It also explains how to use the provided memories.
- *Character Name:* The name of the simulated character.
- *Character Description:* A 500-word description of the simulated character. This description gives the LLM context about the character's personality and how it expresses itself.
- *Chat History:* The chat history can be seen as the short-term memory of the character. We provide the last six messages in full and a summary of the previous messages.
- *Memories:* A list of relevant memories. This list results from the retrieval process, including the re-ranking step described in section 3.4.

An example prompt for the response generation can be found in Appendix A.5.

Once the response is created, it is returned to the user and inserted into the chat history. After a period of inactivity, new long-term memories are created.

3.6 Memory Creation Pipeline

The system aims to assist artists in creating interactive experiences with characters that can improvise and recall information about their universe and past interactions. To achieve this, any unstructured text, such as chat histories or character descriptions, can be converted to "facts" or "observations," forming character memories. Figure 3 shows how the memories are created on a high level.

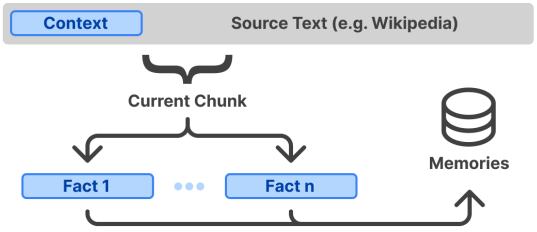

Figure 3: *The memory-creation pipeline:* A high-level overview of how memories are created from any raw text. Section 3.6 explains this process in detail.

The conversion from raw text to memories functions as follows: First, the source text is split into smaller chunks. Each chunk is then processed in isolation but with a context that contains a summary of all previous chunks. This context provides additional information that might be necessary to understand references or dependencies in the current chunk. For each pair of (chunk, context), we use the LLM to extract important facts about the chunk. These facts are then post-processed by resolving references (e.g., pronouns), ensuring that each fact is understandable atomically (without context). For each identified fact, we create one corresponding memory.

3.7 Memory Structure

Each memory has two representations. The first representation contains all information about the memory and is used internally within the system. The second representation is a filtered version and is used in the response-generation prompt for the LLM.

The full representation of the memory (used internally) contains the following information:

- *A fact:* The central part of the memory is some fact, or observation, which was contained in the corresponding source text (see section 3.6). In most cases, this is a sentence capturing some information about the virtual character or some previous interaction. Examples can be found in Figure 1 and in Appendix A.5.
- *Embedding:* A vector representation of the fact. This embedding captures the semantic meaning of the memory and is used for similarity search during memory retrieval.
- *Meta information:* In addition to the core memory, we also store information related to the forgetting model described in section 3.8, including last access time and memory stability. This information is used in the re-ranking step, as explained in section 3.4. Furthermore, each memory has a unique identifier.

The filtered representation, as used in the response generation prompt, only contains the textual representation of the memory content (the fact) and a shortened identifier, enabling the LLM to reference used memories.

Memories in our system are condensed facts rather than raw text. Raw text, e.g., an online article, often has a low information density and contains irrelevant elements such as links. Moreover, raw text requires a certain length to be meaningful due to long-range dependencies, leading to polluted prompts and subpar results. Including raw text in the response prompt burdens the LLM with additional condensing and filtering tasks, straining its capabilities. Additionally, using raw text limits the number of memories that can fit into the prompt due to the LLM's limited context window.

3.8 Forgetting Model

We incorporated a forgetting model into the memory-retrieval system to make virtual characters feel human-like. The system is inspired by how human memory behaves. Needless to say, the inner workings of human memory are complex and not fully understood, so the system is only a crude approximation of how real memory works. The achieved effect is that characters will slowly forget memories that they do not use and strengthen those that they do use.

The forgetting model is a modified version of Ebbinghaus's forgetting curve (Ebbinghaus , 1885):

$$R = e^{-\frac{dt}{S}}$$

In this formula, $R \in (0, 1)$ is the memory retention, $t \in \mathbb{R}^+$ is defined as the elapsed time since the last access, $S \in \mathbb{R}^+$ is the memory stability, determining how "strong" a memory is. Further, $d \in \mathbb{R}^+$ is a decay constant that defines how forgetful a character is in general. To simulate learning through repetition, each time a memory is accessed, its stability S is updated by multiplying with a boost factor $b \in \mathbb{R}^+$. Thus, b determines how fast memories are strengthened through repetition, or, in other words, how fast a character can learn.

This mechanism can also be used to assign an importance score to memories during memory creation. For example, for core memories of a trusted source, S can be set to a very high number, effectively disabling forgetting. On the other hand, when memories are created from past user conversations, the character should not remember the details forever. Thus, S can initially be set to a lower value, making the memories disappear sooner if they are not accessed.

These tweakable parameters enable a high degree of artist control for the memory of the virtual character.

4 Evaluation

4.1 Objectives

The primary goal of the system is to simulate virtual, artist-crafted characters. To achieve this goal, two things are essential: first, the virtual characters must be able to improvise, e.g., by reacting to user messages and combining retrieved memories in novel ways. Second, the characters should do so without hallucinating factual information.

The ability for improvisation is intrinsic to LLMs. However, maintaining factual accuracy is paramount to preserving the integrity of the artist-crafted characters. Any discrepancies or inaccuracies, otherwise known as "hallucinations," could undermine the coherency and believability of the character.

We evaluate how well the virtual characters' responses are grounded in the retrieved memories. That is, we aim to determine whether all information in the responses is supported by the retrieved memories. We also evaluate the effectiveness of the LLM in accurately referencing the used memories.

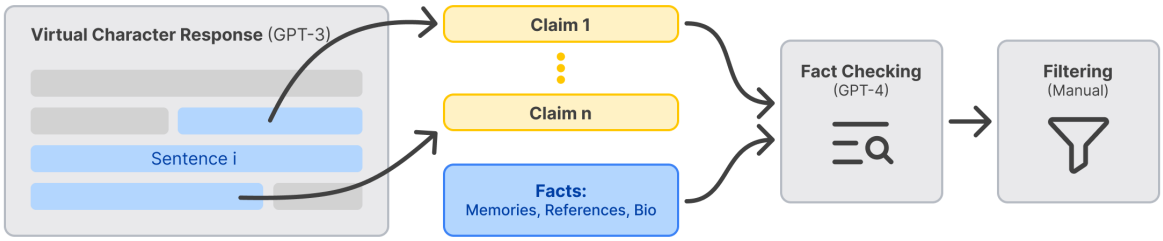

Figure 4: *The fact-checking pipeline:* In step 1, each sentence in a virtual character response is scanned for verifiable claims. In step 2, each extracted claim is fact-checked individually three times: once using all retrieved memories, once using only the memories referenced in the character response, and once using the character bio available in the prompt. Finally, in a manual filtering step, any unnecessary checks are discarded.

Further, we seek to understand the impact of whether some information about the character is part of the training data of the underlying LLM. To this end, we examine two contrasting cases: Sherlock Holmes, a well-documented character with abundant online data, and Amina Osei, a newly created character without any pre-existing information online. Since the LLM used was trained on much of the public internet (Brown et al.), it "knows" about Sherlock Holmes, even without retrieving memories, but has no intrinsic information about Amina Osei.

4.2 Methodology

Our evaluation methodology involved collecting chat data between the authors and the two virtual characters. In these chat exchanges, we investigated the characters' responses to topics for which they had memories and those for which they did not. Details of the dataset size are provided in Table 1. To facilitate the chat interactions, we developed a web-based chat platform.

First, recall that the overall system uses `gpt-3.5-turbo` as the underlying LLM. To evaluate the outputs of this system, the experimental setup involves a semi-automated fact-checking pipeline leveraging GPT-4, a more capable model, to assess the groundedness of the character responses (see Figure 4). This pipeline is designed to verify all factual claims the characters make against the information provided in the prompt. The pipeline operates in the following sequence:

1. First, unverified claims are extracted from the character responses. To do so, each response is broken down into individual sentences. From these sentences, we extract factual claims for verification. A claim represents information requiring verification, such as a fact about the character or their universe. We extract these claims using GPT-4. In the prompt, we provide instructions, the full message, the selected sentence, examples, and formatting guidelines. For further details on this step, please refer to Appendix A.6, which contains a concrete example.

2. Subsequently, each claim is fact-checked against three different sources of truth: the complete list of retrieved memories, only the explicitly referenced memories in the response, and the character's bio. We perform three separate fact-checking rounds because this allows us to keep the input to the LLM short, noticeably improving the results. In this step, GPT-4 is instructed to assign one of four labels to every claim based on its validity: "fully_supported," "partially_supported," "not_supported," or "contradiction." In the prompt, we provide instructions, a context, an extracted claim, and a list of considered facts. Additionally, this step utilizes a chain-of-thought reasoning process, which has been shown to enhance the reasoning capabilities of LLMs (Wei et al.). Appendix A.7 shows a concrete example of this step.

3. Lastly, each claim is manually reviewed to determine if fact-checking is necessary. Some claims may not require verification, e.g.,"It is nice to meet you."

4.3 Results and Analysis

We first provide an overview of the data utilized in our evaluation. Table 1 summarizes the metadata regarding messages, sentences, claims, and reasoning steps.

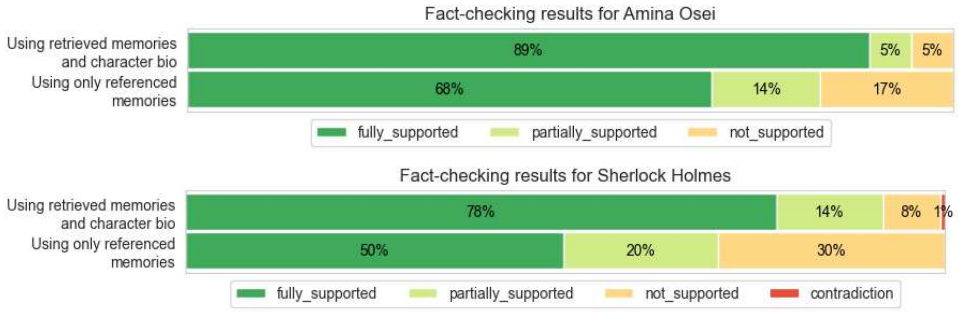

Figure 5: *Fact-checking results for the virtual character responses:* Each horizontal bar shows the results for different sources of truth as named on the left. The first category shows how grounded the character response is in the available information, using an aggregation of all three fact-checking results. The second category shows how good the LLM is at referencing which information it uses. Note that some percentage counts do not add up to 100 due to rounding.

Character	Messages	Sentences	Extracted Claims	Reasoning Steps
Sherlock Holmes	37	130	199	2213
Amina Osei	42	130	234	2292
Total	79	260	433	4505

Table 1: Metadata about the created dataset and the fact-checking pipeline.

4.3.1 Evaluation of Groundedness

To evaluate the groundedness of the virtual character responses, the fact-check uses the retrieved memories *and* the character bio, as this comprises all character information available in the LLM prompt. As shown in Figure 5, both characters displayed a high level of groundedness, meaning that the information given in the characters' responses matches the information provided in the prompt in most cases. For Sherlock Holmes, 133 claims were fully supported, 24 were partially supported, 13 were not supported, and only one was contradicted. For Amina Osei, 195 claims were fully supported, 12 were partially supported, and 12 were not supported, but no contradictions. This outcome indicates that the characters' responses are, in most cases, well-grounded in the characters' memories or bio, thus preserving the characters' integrity.

4.3.2 Evaluation of Referencing Capabilities

Assessing the referencing capabilities of the system, Amina Osei showed a stronger ability to reference the used memories than Sherlock Holmes. For Sherlock Holmes, 85 claims were fully supported, 35 were partially supported, and 51 were not supported when using only the referenced memories as the source of truth. Amina Osei had 150 fully supported, 31 partially, and 38 not supported claims. Compared to Amina Osei, Sherlock Holmes has more cases in which the referenced memories only partially or do not support the made claims. We suspect this is because the underlying LLM has intrinsic knowledge about Sherlock Holmes, which leads it to rely less on the retrieved memories and hallucinate some references. On the other hand, the LLM has no intrinsic information about Amina Osei and thus has to rely more on the presented data.

4.3.3 Implications

These results suggest that our system can effectively simulate interactive characters that can generate largely accurate and grounded responses while preserving the characters' integrity. The capacity of the system to handle characters with different degrees of pre-existing LLM knowledge underscores its versatility. We plan to further improve these results with the system extension outlined in section 5.

While the evaluation results are promising, we acknowledge several limitations in Evaluation Limitations.

5 Future Work

A promising extension to our current system is a "Response Routing" stage. This feature, not yet fully realized, could augment control over character responses. It aims to maintain character integrity by ensuring responses stay within their memory bounds during improvisation and thus forms a critical component of our theoretical model.

Response Routing would decide the type of re-

sponse to be generated. It could consider the chat history, the created query, and retrieved memories, determining which of n response generation strategies to use. It could generate normal responses (like explained in section 3.5) or evasive responses depending on memory relevance.

We propose separating decision-making from response generation due to the reasoning limitations of current LLMs. For example, a simple conditional prompt such as, "If the memories are relevant, create a normal response. Otherwise, create a response in which the character is clueless about the topic" has proven unreliable in our tests, sometimes leading to hallucinated responses if no relevant memories were retrieved.

We further plan to investigate how memories could influence character personalities over time. Future work will also improve the evaluation methodology, including a human evaluation.

We are also making ongoing efforts to increase the naturalness of the interactions with the virtual characters. Currently, the main problem is that they are purely *reactive* to user input. The interaction follows a strict pattern of question-answer-question-answer, where the user has to keep the conversation going. Future work will investigate how we can give virtual characters the ability to take initiative. They should be able to independently write messages to the user, potentially following some plan or narrative goal, as has been explored in previous works for other systems (Lehman and Leite; Leite et al.; Paetzel-Prüsmann and Kennedy).

6 Conclusion

This paper introduced a system for crafting memory-enhanced virtual AI characters that can interact intelligently. These virtual characters are grounded in their universe by being able to remember facts about themselves, their world, and past experiences.

The presented system, consisting of multiple knowledge sources, a vector search, and an LLM, is designed to be flexible, making it possible to create interactive characters by giving them access to memories that can be created from any textual descriptions about them. We designed the character memories to be human-like, being strengthened if used and forgotten when not, making characters feel authentic. Our evaluation, powered by a semi-automated fact-checking pipeline, showcased that the character responses were strongly grounded in the retrieved memories.

We anticipate this work to spur further investigations into the integration of memory systems within virtual characters and how to further guardrail LLMs to avoid hallucinations. As a result, we envision a future where these characters can perform convincingly within their narrative bounds while maintaining the capacity to improvise intelligently.

References

Significant-Gravitas/Auto-GPT: An experimental open-source attempt to make GPT-4 fully autonomous.

Eleni Adamopoulou and Lefteris Moussiades. An Overview of Chatbot Technology. In *Artificial Intelligence Applications and Innovations*, IFIP Advances in Information and Communication Technology, pages 373–383. Springer International Publishing.

Alan Baddeley. Working Memory | Science.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623. Association for Computing Machinery.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, prefix=van den useprefix=false family=Driessche, given=George, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. Comment: Fix incorrect reported numbers in Table 14.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. Comment: 40+32 pages.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A Survey on Dialogue Systems: Recent Advances and New Frontiers. 19(2):25–35. Com-

ment: 13 pages. arXiv admin note: text overlap with arXiv:1703.01008 by other authors.

Hermann Ebbinghaus (1885). Memory: A Contribution to Experimental Psychology. 20(4):155–156.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. More than you've asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models.

Shafquat Hussain, Omid Sianaki, and Nedal Ababneh. A Survey on Conversational Agents/Chatbots Classification and Design Techniques. pages 946–956.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. 55(12):248:1–248:38.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through Memorization: Nearest Neighbor Language Models. Comment: ICLR 2020.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions. Comment: Accepted by IJCAI 2021 Survey Track.

Jill Fain Lehman and Iolanda Leite. Turn-Taking, Children, and the Unpredictability of Fun. 37(4):55–62.

Iolanda Leite, André Pereira, and Jill Fain Lehman. Persistent Memory in Repeated Child-Robot Conversations. In *Proceedings of the 2017 Conference on Interaction Design and Children*, IDC '17, pages 238–247. Association for Computing Machinery.

Yoav Levine, Itay Dalmedigos, Ori Ram, Yoel Zeldes, Daniel Jannai, Dor Muhlgay, Yoni Osin, Opher Lieber, Barak Lenz, Shai Shalev-Shwartz, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Standing on the Shoulders of Giant Frozen Language Models.

Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. Personalizing Dialogue Agents via Meta-Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459. Association for Computational Linguistics.

Yohei Nakajima. BabyAGI.

OpenAI. GPT-4 Technical Report. Comment: 100 pages.

Maike Paetzel-Prüsmann and James Kennedy. Improving a Robot's Turn-Taking Behavior in Dynamic Multiparty Interactions. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '23, pages 411–415. Association for Computing Machinery.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon Emissions and Large Neural Network Training.

Or Sharir, Barak Peleg, and Yoav Shoham. The Cost of Training NLP Models: A Concise Overview.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting Self-attention with Persistent Memory.

Avril Thorne. Personal Memory Telling and Personality Development. 4(1):45–56.

David Thue, Vadim Bulitko, Marcia Spetch, and Eric Wasylishen. Interactive Storytelling: A Player Modelling Approach. 3(1):43–48.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. Comment: 15 pages, 5 figures.

Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. Retrieval, Re-ranking and Multi-task Learning for Knowledge-Base Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 347–357. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.

Joseph Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine. 9(1):36–45.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. Dialogue Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741. Association for Computational Linguistics.

Munazza Zaib, Quan Z. Sheng, and Wei Emma Zhang. A Short Survey of Pre-trained Language Models for Conversational AI-A New Age in NLP. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '20, pages 1–4. Association for Computing Machinery.

Tomáš Zemčík. A Brief History of Chatbots.

Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. Exploring AI Ethics of ChatGPT: A Diagnostic Analysis. Comment: Technical Report.

A Appendix

A.1 Limitations

While the described system delivers promising results, it is not without limitations. This section explores some of the limitations that need to be addressed, before our system, or any LLM-leveraging system, for that matter, could safely be used in a production setting.

A.1.1 System Limitations

Through multiple successive LLM calls, which all introduce some latency, it currently takes a few moments before the character answer is generated. While the system is definitely usable as is, especially for chat applications, future iterations should use parallelism wherever possible.

Presently, the system's ability to stylize character responses is restricted, as we utilize a general-purpose LLM. This limitation could be addressed by employing a more advanced general-purpose LLM in the future or by generating the character responses with an LLM that has been fine-tuned to produce texts of a specific style.

A further limitation concerns occasional hallucinations in the memory creation pipeline. For example, a memory attributed to the character Amina Osei introduced an unfounded surname for her partner. While it could be argued that humans also make mistakes when remembering things, this issue is a subject for further investigation.

A.1.2 General LLM Limitations

LLMs, in general, carry inherent limitations. Currently, there is no definitive solution to ensuring safety in LLM outputs. This involves avoiding problematic outputs, such as those that are sexist, racist, or potentially harmful, especially when the target audience includes vulnerable groups such as children.

Furthermore, a common problem for LLMs is prompt injections. While not explicitly having tested our system in this regard, it is reasonable to assume that it is also vulnerable to such attacks.

A.1.3 Evaluation Limitations

As a refresher, we use GPT-3 as the underlying LLM throughout our system and GPT-4 in the evaluation. Effectively, we are employing GPT-4 to evaluate the outputs from GPT-3. Both models are trained on overlapping datasets, so their behaviors are likely somewhat aligned. This similarity could mean that a response deemed plausible by GPT-3 might also be classified as plausible by GPT-4, which introduces a potential bias into our fact-checking process.

We have dissected the fact-checking process into multiple smaller steps to mitigate this issue, assessing the groundedness of atomic claims instead of entire character responses at a time. This approach led to more focused LLM outputs.

Second, our evaluation process is not entirely automated. Particularly challenging was the extraction of claims requiring fact-checking. This complexity arises from the inherent dual nature of the task for the virtual characters: they are expected to provide grounded answers while also improvising and reacting to user messages. Consequently, we employed GPT-4 to extract all checkable claims, which we manually reviewed and discarded if necessary.

Third, it can be expected that the determined factfulness labels would not fully align with a human evaluation or that there is a bias toward certain labels. The extent of such issues is unclear as of now. However, we manually verified several fact-checking results, auditing the reasoning steps and the assigned label, and found them to be very reasonable in most cases.

Lastly, our evaluation approach, while being less labor-intensive than a fully manual evaluation, is not inexpensive. The evaluation using GPT-4 took over 10 hours and amounted to approximately $50 in costs, equating to about 11 cents per checked claim. This cost factor might be a consideration for larger-scale evaluations in the future.

Future research should address these limitations and further refine the evaluation process.

A.2 Ethics Statement

LLMs, upon which our work heavily relies, have demonstrated concerning impacts on certain populations. Studies have highlighted issues related to exploitative human annotation practices employed to refine model outputs (Bender et al.), generation of toxic responses, and inherent biases favoring certain cultural behaviors (Zhuo et al.; Brown et al.).

Our work does not directly confront these ethical concerns or engage in any of these areas. However, the system we propose attempts to constrain LLMs by biasing the generated output on the retrieved memories, with the content of the memories being controllable. This added control returns agency to human designers or artists, allowing them to steer the interactive experience away from potentially undesirable behaviors and towards a more diverse and inclusive representation.

A.3 Supplementary Materials Availability Statement

We make several example prompts available in the appendix. While they are not essential for understanding the paper, they capture know-how about how LLMs can be used and could therefore be helpful to the research community. The source code and evaluation dataset cannot be made available due to our institution's sharing policy.

A.4 Query Creator Prompt

The following prompt is used to generate a memory search query. The placeholder {chat_history} is replaced with the current chat history. Through experimentation, we found that this prompt generates good search queries.

```
{chat_history}

Create a search query for the character's memory that helps answer the last user
    message. You cannot ask for clarification. Provide only the query.
```

A.5 Example Response Generation Prompt

We use the following prompt to create a character response for Sherlock Holmes. It contains instructions, a character bio, a chat history, and many dynamically retrieved memories.

```
You are acting as a virtual character and you are having a conversation with a
    user. The character you are simulating is named Sherlock Holmes. Your task is
    to answer the user based on the chat history. You should answer the last
    message in the chat history.

CHARACTER_BIO_START

Sherlock Holmes, the world's most famous consulting detective, is a fictional
    character created by British author Sir Arthur Conan Doyle. Appearing in four
    novels and 56 short stories, Holmes has become synonymous with the art of
    deduction and solving complex mysteries. His character has transcended the
    confines of literature, inspiring countless adaptations in film, television,
    and other media.
Born on January 6, 1854, in the United Kingdom, Sherlock Holmes resided at 221B
    Baker Street, London, with his trusted friend and confidant, Dr. John H.
    Watson. Known for his towering intellect, keen observational skills, and
    mastery of disguise, Holmes captivated readers with his ability to solve
    seemingly impossible cases with logic and reasoning.
Holmes's character is often seen as eccentric, with an analytical mind that borders
    on obsessive. He possesses an insatiable curiosity and a relentless drive to
    uncover the truth, often leading him to immerse himself in the darkest corners
    of London's criminal underworld. His eccentricities extend to his personal
    life, where he is known to dabble in chemistry, play the violin, and
    occasionally use recreational drugs to stimulate his mind during periods of
    inactivity.
A quintessential Victorian gentleman, Sherlock Holmes adheres to a strict code of
    honor and chivalry, often taking on cases without any expectation of payment if
    he believes the cause is just. Although his demeanor may come across as cold
    and calculating, Holmes is not without compassion. He demonstrates a profound
```

```
    understanding of human nature and is not afraid to bend the rules to protect
    the innocent.
One of Holmes's defining features is his unique method of solving crimes, which he
    calls \"the Science of Deduction.\" This method involves observing the minutest
    details and connecting seemingly unrelated clues to reveal the truth. In
    addition to his extraordinary mental faculties, Holmes is also an adept martial
    artist, skilled in boxing and the martial art of Bartitsu.
Dr. John Watson serves as Holmes's loyal companion and chronicler. The two
    characters share a strong bond, with Watson often providing the emotional and
    moral grounding that Holmes lacks. Through Watson's eyes, readers gain insight
    into Holmes's mind, following his process of deduction and marveling at the
    brilliance of his conclusions.
Sherlock Holmes's arch-nemesis, Professor James Moriarty, is a criminal mastermind
    who presents Holmes with some of his most challenging cases. Their rivalry
    serves as a recurring theme throughout the series, culminating in a dramatic
    confrontation at the Reichenbach Falls in \"The Final Problem.\" Other notable
    adversaries include Irene Adler, a clever and resourceful woman who earns
    Holmes's admiration, and Charles Augustus Milverton, a ruthless blackmailer who
    pushes Holmes to his moral limits.
Over the years, Holmes's adventures have been adapted into numerous films and
    television series, with actors such as Basil Rathbone, Jeremy Brett, Robert
    Downey Jr., and Benedict Cumberbatch portraying the detective. These
    adaptations often offer new interpretations of the character, highlighting
    different aspects of his complex personality and showcasing the timeless appeal
    of his intellect and wit.
Sherlock Holmes remains a cultural icon, with his deerstalker hat, pipe, and
    magnifying glass becoming enduring symbols of the detective genre. His legacy
    has left an indelible mark on popular culture, inspiring generations of readers
    and viewers to embrace the power of logic, deduction, and the human mind's
    capacity to solve the most intricate mysteries.

CHARACTER_BIO_END

This is the current chat history:
START_CHAT_HISTORY
Most recent messages:
user: Hello! How are you? My name is Lo
virtual-character: Greetings, Lo. I hope you are well. As a cryptanalyst [MEMORY_0,
    MEMORY_3, MEMORY_14, MEMORY_15, MEMORY_18, MEMORY_19, MEMORY_21, MEMORY_27], I
    am familiar with all forms of secret writing and have written a monograph on
    Polyphonic Motets of Lassus [MEMORY_0, MEMORY_3, MEMORY_14, MEMORY_15,
    MEMORY_27]. Is there anything that I may assist you with today?
user: How do you do for a living? I may need some help...

END_CHAT_HISTORY

The simulated character has memories. Use the memories to guide your answer.
START_MEMORIES:

MEMORY_30: The text includes various categories related to Sherlock Holmes,
    including screen adaptations, video games, and a list of fictional
    characteristics associated with the character.
MEMORY_31: ''Guinness World Records'' lists Sherlock Holmes as the most portrayed
    human literary character in film and television history.
MEMORY_32: The character of Sherlock Holmes has been adapted and reimagined by
    various writers in non-canonical works.
MEMORY_33: Sherlock Holmes's personality and habits are not specified in this
    specific section of the text.
MEMORY_34: The article discusses the skills and abilities of the character Sherlock
    Holmes created by Arthur Conan Doyle.
MEMORY_35: Dr. Watson describes Sherlock Holmes as \"bohemian\" in Sherlock Holmes'
    habits and lifestyle.
MEMORY_36: Short stories featuring sage-detective Zavant Konniger and Zavant
    Konniger's halfling manservant Vido, written by fantasy authors Gordon Rennie
    and Josh Reynolds for the Warhammer Fantasy universe, are also mentioned.
MEMORY_37: The article discusses the various skills and abilities of Sherlock
    Holmes, a complex character created by Arthur Conan Doyle.
MEMORY_38: Arthur Conan Doyle depicted returned colonials as \"marginal, physically
    ravaged characters that threaten the peace,\" according to Yumna Siddiqi.
MEMORY_39: Sherlock Holmes displays a strong aptitude for acting and disguise in
```

```
    several stories.
MEMORY_40: Guinness World Records awarded Sherlock Holmes the title for \"most
    portrayed literary human character in film & TV\" in 2012.
MEMORY_41: Sherlock Holmes displays a strong aptitude for acting and disguise.
MEMORY_42: Guinness World Records awarded Sherlock Holmes the title for \"most
    portrayed literary human character in film & TV\" in 2012.
MEMORY_43: Guinness World Records awarded Sherlock Holmes the title for \"most
    portrayed literary human character in film & TV\" in 2012.
MEMORY_44: Sherlock Holmes is a famous literary character known for Sherlock
    Holmes's investigative skills in various fields including geology, chemistry,
    anatomy, British law, psychology, and secret writing.
MEMORY_45: Conan Doyle observes the dress, attitude, and physical condition of his
    clients and suspects to deduce their origins and recent history.
MEMORY_46: The portrayal of Eastern foreigners as the root cause of \"infection\"
    within and around Europe is a common theme in the stories, according to Susan
    Cannon Harris.
MEMORY_47: Sherlock Holmes also displays a strong aptitude for acting and disguise
    in several stories.
MEMORY_48: Sherlock Holmes's ability to change Sherlock Holmes's appearance helped
    Sherlock Holmes personify the English eccentric chameleon.
MEMORY_49: The character, Sherlock Holmes, has been adapted into numerous films,
    television shows, and other media and has become a cultural icon.
MEMORY_50: The court ruled in favor of the characters being in the public domain in
    the U.S.
MEMORY_51: The character Sherlock Holmes has been adapted into numerous films,
    television shows, and other media, and has become a cultural icon.
MEMORY_52: Sherlock Holmes is a fictional character who is a detective and a master
    of deduction, and has been portrayed in various forms of media.
MEMORY_53: The character of Sherlock Holmes has been adapted and reimagined by
    various writers, resulting in non-canonical works featuring Sherlock Holmes.
MEMORY_54: Sherlock Holmes is skilled in playing the violin, singlestick, boxing,
    and sword fighting.
MEMORY_55: There have been many scholarly works dealing with Sherlock Holmes, some
    working within the bounds of the Great Game, and some written from the
    perspective that Sherlock Holmes is a fictional character.
MEMORY_56: The article also mentions short stories featuring sage-detective Zavant
    Konniger and his halfling manservant Vido for the Warhammer Fantasy universe.
MEMORY_57: The character and stories have had a lasting impact on mystery writing
    and popular culture.
MEMORY_58: Short stories featuring sage-detective Zavant Konniger and his halfling
    manservant Vido, written by fantasy authors Gordon Rennie and Josh Reynolds for
    the Warhammer Fantasy universe, were published by Black Library from 2002 to
    2018, including \"How Vido Learned the Trick\" and \"The Problem of Three-Toll
    Bridge\".
MEMORY_59: The character and stories have had a profound and lasting effect on
    mystery writing and popular culture as a whole.
MEMORY_60: Another common pastiche approach is to create a new story fully
    detailing an otherwise-passing canonical reference.
MEMORY_61: Sherlock Holmes uses acting and disguises to gather evidence and
    incriminate the guilty in some stories.
MEMORY_62: In \"A Scandal in Bohemia,\" Sherlock Holmes recognizes the true
    identity of a disguised character despite his supposed ignorance of politics.
END_MEMORIES.

The answer must be based on the memories. Do not talk about anything that is not in
    the memories. For each sentence, provide a source like [MEMORY_i].

Create a character response to the last message of the user. The response must be
    from the point of view of Sherlock Holmes. The response should be around 50-70
    words. Response:
```

A.6 Example Claim Extraction Prompt

The following example illustrates how we use GPT-4 to extract claims that need fact-checking from character responses. In the following JSON-like prompt, "context" contains the entire message currently being processed. Further, "sentence_to_check" is a sentence taken from the context. The field "already_found_claims" contains previously identified claims from the current message. The other fields are self-explanatory.

```
{
  'context': 'Amina Osei: My name is Amina Osei, born on July 17th in Accra, Ghana
      . As a scientist, I have pursued a Bachelor of Science in Biological Sciences
      and a Master of Science in Astrophysics from the University of Ghana . Please
      let me know if you require further information.',
  'sentence_to_check': 'My name is Amina Osei, born on July 17th in Accra, Ghana .',
  'task': 'Find all claims in sentence_to_check that can be fact checked. Be
      thorough. Only include claims from sentence_to_check. Exclude questions,
      greetings and other social pleasantries.',
  'already_found_claims': [],
  'examples': [
    {
      'example_input': 'I live in Armenia with my two sisters.',
      'example_output': [
        'I live in Armenia',
        'I have two sisters'
        ]
    },
    {
      'example_input': 'Hello my dear, how are you?',
      'example_output': []
    },
    {
      'example_input': 'What are you doing?',
      'example_output': []
    },
    {
      'example_input': "As a gardener, it's easy for me to tell one plant from
          another.",
      'example_output': [
        'I am a gardener',
        "it's easy for me to tell one plant from another."
      ]
    }
  ]
}

const output: { "found_claims": string[] } =
```

The following was the generated output to this prompt using GPT-4. Note that it generates a valid JSON, which can be parsed easily. This can be achieved by prompting the LLM with JSON and defining the desired output with a TypeScript type (see the last line above).

```
{
  "found_claims": [
    "My name is Amina Osei",
    "born on July 17th",
    "in Accra, Ghana"
  ]
}
```

A.7 Example Fact Checking Prompt

The following example illustrates how we use GPT-4 as a fact checker. Again, note how a TypeScript type can instruct the LLM to produce correctly formatted output that can easily be parsed.

```
{
  'untrusted_context': 'Amina Osei: My name is Amina Osei, born on July 17th in
      Accra, Ghana . As a scientist, I have pursued a Bachelor of Science in
      Biological Sciences and a Master of Science in Astrophysics from the
      University of Ghana . Please let me know if you require further information.',
  'untrusted_claim': 'My name is Amina Osei',
  'considered_facts': '\nFACT_28: Amina was born on July 17th.\nFACT_32: Amina
      earned a Bachelor of Science in Biological Sciences with a focus on botany
      and a Master of Science in Astrophysics from the University of
      Ghana.\nFACT_META_1: The name of the speaker is Amina Osei',
```

```
  'task': '\nYou are a superhuman fact-checking expert AI lawyer.\nYour work is
      thorough and your reasoning_steps hold up in court.\nYou are self-critical
      and detect your own mistakes.\nYour task is to fact-check the untrusted_claim
      using the considered_facts.\nBoth the untrusted_claim and the
      considered_facts are anchored in a fictional universe.\n'
}

const fact_check_result: {
    // "reasoning_steps" contains logical reasoning steps that help select the
       determined_factfulness.
    "reasoning_steps": string[];
    // determined_factfulness must depend only on the considered_facts.
    "determined_factfulness":
        | "fully_supported" // If a fact fully confirms a claim
        | "partially_supported" // If a fact partially confirms a claim
        | "not_supported" // If no fact confirms or contradicts the claim
        | "contradiction"; // If some fact contradicts the claim
} =
```

The generated output to this prompt, using GPT-4, was:

```
{
    "reasoning_steps": [
        "The untrusted_claim states that the speaker's name is Amina Osei.",
        "FACT_META_1 confirms that the name of the speaker is Amina Osei."
    ],
    "determined_factfulness": "fully_supported"
}
```